Towards Self-Adaptive Management Frameworks: the Case of Aggregated Information Monitoring

Rafik Makhloufi*, Guillaume Doyen*, Grégory Bonnet† and Dominique Gaïti*
*ICD/ERA, UMR 6279. Université de Technologie de Troyes
Email: {rafik.makhloufi, guillaume.doyen, dominique.gaiti}@utt.fr
†GREYC/MAD UMR 6072. Université de Caen Basse-Normandie
Email: gregory.bonnet@unicaen.fr

***Abstract*—In this paper, we propose a decentralized self-adaptive management framework that infers the current context of the hosting environment and dynamically decides what management strategy should be used in order to minimize the cost and maximize the performance of management. We investigate the use of this framework in the case of the aggregated information monitoring, where only one scheme (i.e. situated or global) is generally used for collecting aggregated information whatever the nature of the management information and its environment are. We propose a self-adaptive aggregation mechanism based on fuzzy logic, that estimates the cost and the performance of given aggregation schemes.**

***Index Terms*—Autonomous Networking, Decentralized Aggregation, Management Information, Fuzzy Logic.**

I. INTRODUCTION

Current networks and services constantly evolve in terms of complexity, size, distribution and dynamics. During the last decade, numerous approaches have emerged in order to manage them in a decentralized and autonomous way. These autonomous approaches are deployed on heterogenous and dynamic environments where it is crucial for these approaches to adapt themselves to the supporting network operational behavior. For example, in mobile ad-hoc networks (MANETs) and peer-to-peer (P2P) ones, the network size is continuously evolving due to, respectively, nodes mobility and peers churn. Consequently, the management approach should adapt itself to such dynamics and operational states.

In this paper, we propose a decentralized self-adaptive management framework that infers the current operational state of the network and dynamically decides what management strategy should be used according to this context. We investigate the use of this framework in the case of the aggregation of management information, where we propose a self-adaptive aggregation mechanism based on fuzzy logic, that estimates the cost and the performance of given aggregation schemes, computes their utilities and then decides collectively what is the best aggregation scheme to use. We focus on the utility evaluation, which is the core of the architecture dedicated to information analysis for the decision-making, and then we validate the developed fuzzy-based model.

The reminder of this paper is organized as follows. In Section 2, we present the related work. We present the functional elements of the proposed adaptive management framework in Section 3. Then, we present the utilization of this approach for an adaptive aggregation in Section 4. Subsequently, we give early validation results of this approach in Section 5. Finally, we conclude this work and we present our perspectives.

II. RELATED WORK

Autonomous management systems [1] need to collect local and global management information in order to analyze them and to make management decisions. Several research efforts propose to collect it through aggregation schemes.

In [2] and [3], the authors propose and evaluate some gossip- and tree-based aggregation schemes. [2] shows that each category outperforms the other one for a given performance criteria, while [3] shows that the tree outperforms gossip for all the studied criteria. Unfortunately, these studies do not discuss the situated aggregation schemes.

In [4] and [5], we gave an overview of a set of decentralized aggregation schemes and a multi-criteria evaluation of them, showing that each aggregation scheme is better than the others for a given evaluation criteria. [5] shows that global schemes (i.e. gossip and tree) provide a good accuracy at the cost of a larger convergence time and overhead, while situated approaches minimize their cost and provide a high reactivity to the Autonomic Managers (AMs) with a low accuracy.

Globally, these studies address decentralized aggregation schemes and propose to evaluate their cost and performance through simulations or algorithmic complexity. They show that aggregation strategies behave differently depending on the network operational state and the management information behavior. Through these studies, we observe that it is crucial to establish an adaptive approach that adapts itself to the supporting network operational behavior in order to be able to design efficient management systems. But, existing systems propose to use one aggregation scheme in a static and ad-hoc way without taking into account the nature of management information and the one of its environment.

III. SELF-ADAPTIVE MANAGEMENT ARCHITECTURE

In order to be able to design efficient frameworks that minimize the cost and maximize the performance of management, we propose to use a self-adaptive approach that infers the current network operational state and dynamically decides according to this one what management strategy should be used and/or what changes must occur on it. This framework takes into account management policies through user preferences representing, for example, a tradeoff between performance and cost of the management process.

As depicted in Figure 1, an autonomous system is composed of a set of interactive AMs that manage their internal behavior and their relationships with other AMs. The control loop of autonomic computing consists of four main functions which are Monitor, Analyze, Plan, Execute (MAPE) [6]. This model can be applied to the adaptive configuration of the management plane with four components that enable a dynamically selection of a management approach according to the current context. The global architecture of the resulting framework is composed of:

- Network state monitor: this process is responsible for the monitoring and data-gathering about the current operational state of the network (e.g. size, dynamics, neighborhood degree). It uses sensors to collect data from local or from other AMs and then it sends them to the next component for analysis. An aggregation scheme can be used to collect this management information.
- Utility evaluator: based on the data provided by the network state monitoring engine, the utility evaluator analyzes them, estimates the values cost and the performance of given management strategies. According to these values, a utility function is then used for the computation of a pertinence metric that shows how much a management strategy is relevant. This utility function allows a tradeoff between performance and cost.
- Collaborative decision-making: according to the given utility of each management strategy, this component decides if the approach having the best utility is more relevant to be used than the current and what changes should be applied on the selected approach. To do this, each AM collaborates with other AMs in the network in order to make a collective decision on a possible change so that they minimize the cost and maximize the performance of management.
- Change operation engine: once the changes that should have occurred are defined by the collaborative decision-making mechanism, the change operation component is in charge of reconfiguring the management plane apply the given decisions. This requires synchronized and rapid actions through all AMs so that the change is seamless.

IV. SAAM: A SELF-ADAPTIVE AGGREGATION MECHANISM

In this section, we present a use case of the previous architecture in the context of aggregated information monitoring. Indeed, an autonomous management system need to collect management information from the network elements. This information can be collected through aggregation schemes. Aggregation [4] is intended as a summarizing mechanism of the overall state within the network. For a best aggregation quality, we design a new decentralized self-adaptive aggregation framework (SAAM), based on the previously described architecture. According to network and information characteristics, each decentralized entity of the SAAM computes the utility of each aggregation scheme and then, with the help of the other elements, it decides what is the best aggregation scheme to use.

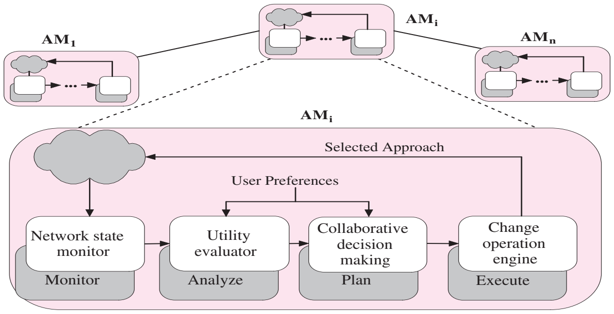

Fig. 1. Global Architecture of the Self-Adaptive Aggregation Framework.

A. Considered Aggregation Schemes

We consider typical situated and global aggregation schemes that we benchmarked in [5]. These are:

- Gossip (G): a push-pull gossip scheme inspired from [7], where in each round, each node contacts one neighbor chosen uniformly at random and exchanges information with it. Thus, a global aggregate will be computed and made available across all the network nodes.
- Tree (T): a push tree-based scheme based on GAP [8], where nodes communicate their local aggregates to a single root node that computes an overall aggregate and spreads it on all the interested nodes through a publish-subscribe mechanism.
- Situated view: a pull-based situated scheme based on HyParView [9] and Propagate2All [2], where each node maintains a view of its h-hops neighbors. Thus, each node collects information from its view and then computes a partial aggregate. We consider a one-hop situated view (SV1) and a two-hops situated view (SV2).

B. Utility Evaluator

In this study, we focus on the utility evaluator. As illustrated in Figure 2, the utility evaluator is composed of Fuzzy Logic Controllers (FLCs) and a utility function that can be computed, for example, through MADM (Multiple Attribute Decision Making) [10]. The latter uses the numerical values of the cost and the performance provided by the FLCs for the computation of a pertinence metric that shows how much an aggregation scheme is relevant to be used.

We rely on fuzzy logic because it provides a formal methodology for representing, manipulating, and implementing a human's heuristic knowledge about how to control a system. It allows a performant decision-making under incomplete and uncertain information, here, about the network operational state. Moreover, computing with fuzzy sets is a necessity when

(1) the available information is too imprecise to justify the use of numbers, and (2) there is a tolerance for imprecision which can be exploited to achieve tractability, robustness, low solution cost, and better rapport with reality [11], [12]. We propose here to use two fuzzy logic controllers to process the inputs on the network state in order to estimate, respectively, the performance and the cost of each aggregation approach. These FLCs operates according the following main steps: fuzzification, inference and defuzzification.

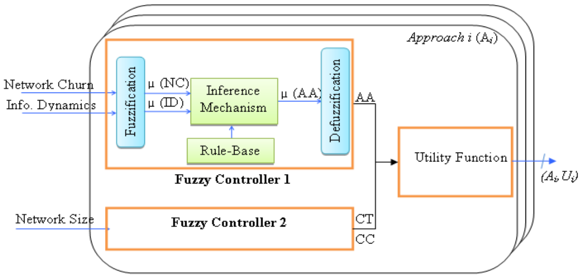

Fig. 2. Architecture of the Utility Evaluator.

1) Fuzzification: According our previous evaluation studies [5], [13], we distinguish, on one hand, two main factors that impact the performance of an aggregation scheme: network and information dynamics. On the other hand, the cost of an aggregation scheme depends on the network size. In order to consider them as inputs of the FLC, we need to convert these three parameters into fuzzy inputs presenting different degrees of membership so that they can be interpreted and evaluated in an inference mechanism. These factors are defined as follows:

- Network size (NS) [13]: it is the number of nodes in the network. This factor directly affects the cost of aggregation schemes.
- Network dynamics (ND) [5]: it consists in nodes arrivals in the system and their departures that can be planned or due to a sudden failure. We characterize it by the sum of both nodes arrival and departure rates.
- Information dynamics (ID) [5]: it determines the evolution degree of a management information over the time. We characterize it through the changing degree (i.e. distance) between two consecutive values of the monitored variable on the managed element.

Since in the literature, the most acknowledged studied cost and performance criteria for aggregation schemes are convergence time, communication cost and accuracy, we choose them as outputs of these FLCs. They are defined as follows:

- Convergence Time (CT) [13]: it is the necessary time between the initialization of the aggregation process and the time t when all nodes hold the aggregation results.
- Communication Cost (CC) [13]: it is the sum of sizes of messages sent between any node pairs (i, j) during the aggregation process. Since all messages have approximatively the same size, the communication cost is measured by the number of messages sent by all nodes.
- Aggregation Accuracy (AA) [5]: it is the precision level of the aggregation scheme when computing aggregates. It is measured by the average deviation (i.e. distance) between the estimated aggregates and the actual global aggregate that should be computed.

2) Inference: In order to evaluate which control rules have to be activated at the current time according to the inputs and then decide what the output should be, each FLC relies on an inference mechanism. The latter uses a set of rules, on the form "**If** premise **Then** consequent" to estimate the cost and the performance of each aggregation scheme. We define three membership functions for each FLC variable in the rule-base: low, medium and high. Table I and Table II below illustrate, respectively, the rules used by the FLCs to estimate both the performance and the cost of each aggregation scheme. For example, in the case of gossip on Table I: **If** ND=Medium **and** ID=Low **Then** AA=Medium.

TABLE I
RULES OF THE FIRST FLC.

		Gossip			**Tree**			**SV1**			**SV2**		
		ND			**ND**			**ND**			**ND**		
		L	**M**	**H**	**L**	**M**	**H**	**L**	**M**	**H**	**L**	**M**	**H**
ID	**L**	H	M	L	H	L	L	L	L	L	L	L	L
	M	H	M	L	H	L	L	L	L	L	L	L	L
	H	M	L	L	M	L	L	L	L	L	L	L	L

TABLE II
RULES OF THE SECOND FLC.

		Gossip		**Tree**		**SV1**		**SV2**	
		CT	**CC**	**CT**	**CC**	**CT**	**CC**	**CT**	**CC**
NS	**L**	M	L	M	L	L	L	L	L
	M	H	L	M	L	L	L	L	M
	H	H	M	M	L	L	M	L	H

3) Defuzzification: The defuzzification interface converts the conclusions reached by the inference mechanism (i.e. fuzzy sets) into quantifiable outputs. In the case of SAAM, the defuzzification component will converts the fuzzy measures on the cost and the performance of each aggregation approach into numeric values that will be used by the utility function. To do this, we consider the center of gravity which is the most standard defuzzification method [14]. The latter computes the center of gravity of the area under the membership functions of the inference fuzzy outputs.

V. EARLY VALIDATION RESULTS

In this section, we give early validation results of the proposed adaptive approach, focusing on the FLCs of the utility evaluator. This aims at checking if these FLCs enable the AMs to consistently estimate by themselves, in a decentralized way, the cost and the performance of each aggregation scheme.

A. Evaluation Setup

In order to show the influence of dynamics and network topology on the cost and the performance of an aggregation scheme, we rely on the fuzzy logic toolbox of Matlab [15], in which we implement the two FLCs. We propose to characterize their inputs and outputs according to the performance evaluation results provided in [5] for each aggregation scheme

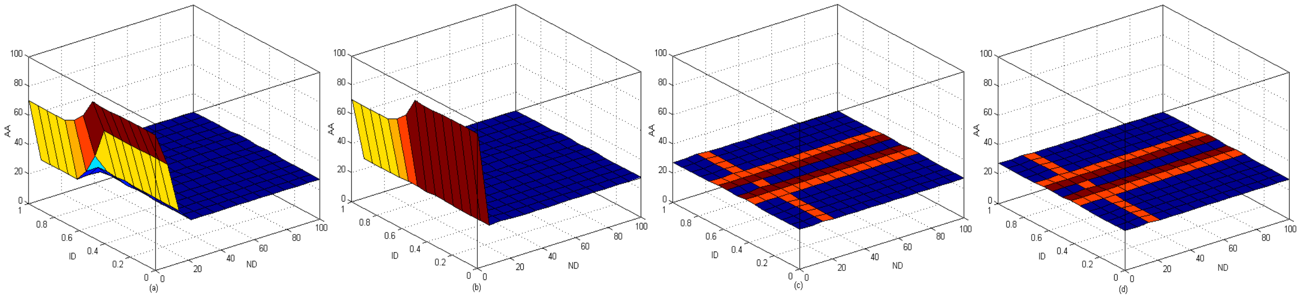

Fig. 3. Aggregation Accuracy for (a) Gossip, (b) Tree, (c) SV1, and (d) SV2 According to Network and Information Dynamics.

(i.e. gossip, tree, SV1 and SV2). The fuzzification of these variables is shown in Figure 4, where we use normalized values for both dynamics and churn. Thus, ND and AA are ranging in [0;100] while ID∈[0;1].

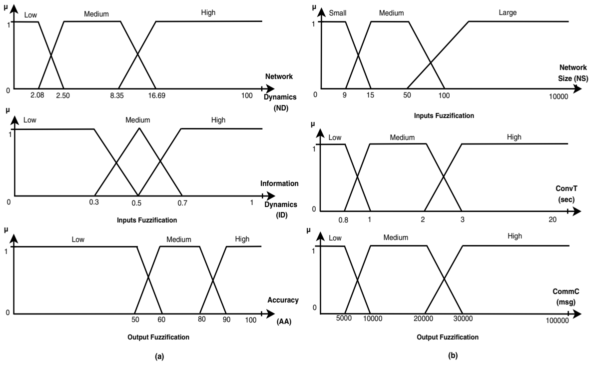

Fig. 4. Fuzzification of the FLCs inputs and outputs: (a) FLC1, (b) FLC2.

B. Results

Figures 3 and 5 display respectively how aggregation performance and cost depend on the FLC inputs, i.e., network churn, information dynamics and network size. Results are direct outputs of the fuzzy controller. They provide an abstract view of the relative performance of the given schemes under different conditions. We observe in Figure 3 that, for both gossip and tree, the aggregation accuracy decreases with an increase of network and information dynamics. We also observe that tree is more affected by the network dynamics than gossip. Concerning SV1 and SV2, they ensure a low accuracy whatever the level of dynamics is. We notice in Figure 5.a a high convergence time that increases when having a large number of nodes in the case gossip. This convergence is medium for tree and low for SV1 and SV2, independently of the network size. In Figure 5.b, we see that the communication cost increases according to the network size. Especially, we observe a high communication cost for both gossip and SV2 followed by SV1 and then a low communication cost for tree. However, for example in Figure 5.a, convergence time on the tree is constant while it should have a logarithmic distribution according to the network size (NS). In our prior simulations [5], CT is proportional to the network size. Globally these results confirm our expectations and they are coherent with those obtained in [5] through simulations. This enables each node to consistently estimate, in a decentralized way, the cost and the performance of each aggregation scheme.

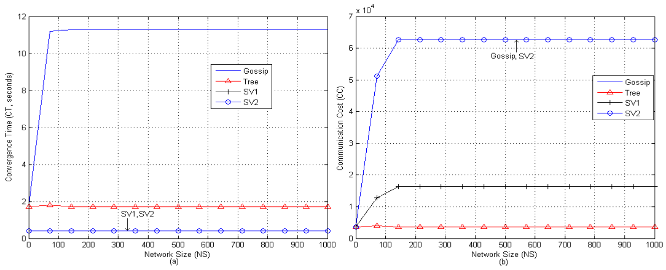

Fig. 5. Estimated Convergence Time (a) and Communication Cost (b) According to the Network Size.

VI. CONCLUSION AND PERSPECTIVES

In order to be able to design efficient management frameworks, we propose in this paper to use a self-adaptive architecture that adapts itself to the supporting network operational behavior. We instantiate this architecture in the case of aggregation, where it is crucial to efficiently provide the required management information to the decision-making process while minimizing its cost and maximizing its performance. In order to estimate them, we used a fuzzy approach and especially two fuzzy controllers that consider network and information dynamics and network size. Validation results confirm the existing simulation ones, enabling an autonomous manager to get a consistent estimation of the cost and the performance of each aggregation scheme. This work opens numerous perspectives. In order to enhance the characterization of the aggregation schemes and the validation results, we are currently working on the enhancement of the utility evaluator with the definition of a consistent utility function. We are also working on the next functional blocs that present strong challenges: the establishment of a global agreement on the best strategy to use, and the rapid and seamless reconfiguration of the autonomous management plane. It is also important to address the adaptation cost of SAAM after implementing all the blocs. At that moment, realistic situations and use cases (e.g. P2P VoD streaming, MANeTs) can be studied by choosing the suitable parameters (e.g. weights of the utility function), allowing a tradeoff between cost and gain.

REFERENCES

[1] P. Horn, “Autonomic computing: IBM’s Perspective on the State of Information Technology,” 2001.

[2] M. Bawa, H. Garcia-Molina, A. Gionis, and R. Motwani, “Estimating aggregates on a Peer-to-Peer network,” Stanford InfoLab, Tech. Rep., 2003.

[3] F. Wuhib, M. Dam, R. Stadler, and A. Clem, “Robust monitoring of network-wide aggregates through gossiping,” *IEEE Transactions on Network and Service Management*, vol. 6, no. 2, pp. 95–109, 2009.

[4] R. Makhloufi, G. Bonnet, G. Doyen, and D. Gaïti, “Decentralized Aggregation Protocols in Peer-to-Peer Networks: a Survey,” in *IEEE International Workshop on Modelling Autonomic Communications Environments*, Venice, ITALY, 2009, pp. 111–116.

[5] R. Makhloufi, G. Doyen, G. Bonnet, and D. Gaïti, “Impact of Dynamics on Situated and Global Aggregation Schemes,” in *IFIP International Conference on Autonomous Infrastructure, Management, and Security*, Nancy, France, 2011, pp. 148–159.

[6] J. O. Kephart and D. M. Chess, “The Vision of Autonomic Computing,” *Computer*, vol. 36, no. 1, pp. 41–50, 2003.

[7] M. Jelasity, A. Montresor, and O. Babaoglu, “Gossip-based aggregation in large dynamic networks,” *ACM Transactions on Computer Systems*, vol. 23, no. 3, pp. 219–252, 2005.

[8] M. Dam and R. Stadler, “A generic protocol for network state aggregation,” in *Radiovetenskap och Kommunikation*, 2005.

[9] J. Leitao, J. Pereira, and L. Rodrigues, “Large-Scale Peer-to-Peer Autonomic Monitoring,” in *IEEE DANMS*, 2008, pp. 1–5.

[10] C.-L. Hwang and K. Yoon, *Multiple Attribute Decision Making*. Springer-Verlag, 1981.

[11] L. Zadeh, “Fuzzy logic = computing with words,” *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 2, pp. 103–111, 1996.

[12] K. M. Passino and S. Yurkovich, *Fuzzy Control*. Addison Wesley Longman, 1998.

[13] R. Makhloufi, G. Doyen, G. Bonnet, and D. Gaïti, “Situated vs. Global Aggregation Schemes for Autonomous Management Systems,” in *IFIP/IEEE International Workshop on Distributed Autonomous Network Management Systems*, Dublin, Ireland, 2011, pp. 1131–1135.

[14] W. V. Leekwijck and E. E. Kerre, “Defuzzification: criteria and classification,” *Fuzzy Sets and Systems*, vol. 108, no. 2, pp. 159–178, 1999.

[15] MathWorks, “Matlab - the language of technical computing,” 2011, http://www.mathworks.com/products/matlab/.